

# Videojuego retro

Programación y diseño hardware de un videojuego

Proyecto final de carrera

Autor: Ricardo Gil González

Director: Manuel Lamich Arocas

Titulación: Grado en Ingeniería Electrónica Industrial y Automática

Departamento: Departamento de Ingeniería Electrónica

Fecha de entrega: 09/06/2016

## Índice

Introducción .....	4
Objetivos.....	5
Herramientas utilizadas.....	6
Hardware .....	6
Información de las placas Arduino.....	6
Información modulo bluetooth.....	7
Información pantalla .....	9
Información zumbador.....	11
Información placa PCB LCD-DUE .....	11
Caja .....	12
Mando .....	12
Software.....	13
Programas informáticos para el código del juego .....	13
Programas informáticos para el código del mando .....	14
Descripción del proyecto.....	15
Hardware .....	15
Software.....	17
Resultados .....	20
Conclusiones.....	22
Futuras ampliaciones.....	25
Apéndices .....	26
Cronología del desarrollo del programa .....	26
Librería UTFT .....	31
Instalar la pantalla con cables.....	32
Programa comentado .....	34
Programa comentado mando.....	45
Presupuesto material.....	47
Bibliografía.....	48

## Listado figuras y tablas

Tabla 1 Características placas Arduino.....	7
Tabla 2 Clasificación bluetooth según el alcance .....	8
Tabla 3 Clasificación según capacidad de canal.....	8
Tabla 4 Relación pines pantalla y Arduino.....	33
Tabla 5 Precio elementos producto .....	47
Figura 1 Diseño hardware proyecto.....	15
Figura 2 Diagrama de bloques .....	17
Figura 3 Direcciones pelota .....	28
Figura 4 Sentido del rebote de la pelota.....	28
Figura 5 Interfaz programa Appinventor.....	45
Figura 6 Código mando.....	46

## Introducció

A lo largo de la carrera se ha aprendido mucha teoría de varios ámbitos del sector industrial ya sea la creación de circuiterías mediante cálculos y estudios de viabilidad averiguando que opción es la mejor, a poder diseñar una pieza tanto en 2D o 3D y tener una mente entrenada en la visualización de elementos en 3D para poder entender esquemas, croquis y planos técnicos y saber utilizar programas informáticos de diseño de piezas, pasando por la programación de robots en diferentes lenguajes y ámbitos industriales o la creación de interfaces para un cliente. También se ha asimilado de la parte de administración de empresas como organizar la producción.

El proyecto de final de grado debe ser un trabajo que muestre todo ese conocimiento basándose en la realización de un trabajo individual. Con los conocimientos adquiridos se podrá entender los datasheets de los componentes que se vayan a utilizar y el funcionamiento interno, es decir, saber cómo trabaja, qué función tiene cada pin de entrada o salida. De la parte del software se aplicarán todos los conocimientos de las asignaturas impartidas, que son, por ejemplo diversos lenguajes y se han manejado multitud de programas informáticos. Otro bloque importante dentro del mundo de la industria es la parte de diseño, que se ha estudiado en una asignatura donde se aprende a trabajar con un programa informático capaz de dibujar en 2D o 3D y hacer planos. Se querrá trasladar esos conocimientos a un ejemplo práctico.

No tan solo conocimientos de teorías se han instruido en estos años también se ha asimilado en la multitud de informes de laboratorio y en la asignatura de metodología de proyectos como planificarse, dividir el trabajo, los tiempos y como redactar un informe en el ámbito de la ingeniería. Una competencia transversal que se profundiza a lo largo de la carrera es la del trabajo y el aprendizaje autónomo, en este proyecto se deberá trabajar con elementos electrónicos con los que no se estaban familiarizados y programas informáticos de desarrollo de programas diferentes con nuevos lenguajes y que se deberá buscar la información y aprender rápido.

En resumen, este proyecto ayudará a consolidar los conocimientos adquiridos y a hacer una visión general de todo lo impartido en la carrera complementándolo con los nuevos conocimientos que se aprenden después del proyecto.



## Objetivos

El videojuego Pong de los años 70 fue de las primeras consolas en comercializarse y revolucionó el mundo del entretenimiento. El juego simula el tenis mesa y consiste en la visualización de dos raquetas en forma rectangular que se pueden mover en vertical en sus ejes correspondientes y de una pelota rectangular que rebotaba en los márgenes de la pantalla. El usuario podía ver un marcador que se incrementaba cuando una de las dos raquetas no impactaba con la pelota.

El objetivo del proyecto final de carrera es el de recrear este videojuego con componentes electrónicos del siglo XXI, éstos sustituirán a la circuitería ya que la consola original se basaba en transistores, resistencias, condensadores, etc. Otra parte que formaba parte de la consola era el mando que utilizaba potenciómetros para mover la raqueta, en este punto el objetivo es el de crear un mando con el mismo concepto de la utilización de los potenciómetros por lo que se deberá buscar todas las piezas necesarias y crear un mando ergonómico para que sea cómodo de utilizar. La consola original no incorporaba pantalla ya que se conectaba a la televisión, pero para este proyecto se incorporará para que sea portátil, es decir, se pueda jugar donde se desee.

Es un gran proyecto con muchas partes que se deben desarrollar en paralelo para cumplir los plazos, es por esa razón que se deberán poner en práctica todos los mecanismos que se han ido aprendiendo a lo largo de la carrera de cómo se deben planificar los trabajos. Se deberán planificar los tiempos de duración de cada apartado. Otro recurso aprendido es el de dividir el problema en partes más pequeñas para facilitar su solución. La habilidad más utilizada y desarrollada en toda la carrera es la del aprendizaje autónomo y que ése sea lo más rápido posible, habilidad que también se pondrá en práctica ya que se utilizarán componentes electrónicos, lenguajes informáticos que serán desconocidos y se deberá buscar, aprender y asimilar con rapidez la información necesaria para su utilización.

## Herramientas utilizadas

### Hardware

#### Información de las placas Arduino

Hay multitud de tipos de placas Arduino según el grado de conocimiento de la persona y para que las quieras utilizar. A la hora de comprar se deben mirar las características de cada placa, dividiéndose éstas según su nivel de complejidad; la primera que se encuentra es Arduino UNO, 101 y PRO, éstas se caracterizan por su fácil manejo desde el primer momento ya que solo se necesita conectarlos y cargar el programa, están pensados para los que se inician en la utilización de estos dispositivos. Para facilitar el aprendizaje se puede comprar un kit con 15 tutoriales donde el cliente podrá practicar e ir aumentando su nivel de comprensión progresivamente con ejercicios cada vez más complejos.

El segundo grupo que se encuentra tiene las mismas características que el primero pero mejoradas, es decir, son placas con mejores cualidades orientadas a un cliente que tiene unos conocimientos previos y necesita hacer un proyecto de más complejidad. Las cualidades que más destacan es el aumento de la frecuencia de transmisión de datos, el aumento de los pines y el de la memoria. Estas placas se llaman Arduino DUE, MEGA y ZERO.

El tercer grupo se basa en si el cliente necesita conectar un dispositivo a internet. Para ello se dispone de Arduino YÚN. La característica diferenciadora es que se puede comunicar con un ordenador que contenga un sistema operativo LINUX.

A continuación se muestra en la tabla 1 las características de una placa de cada grupo con lo que se puede hacer una comparativa de cualidades.

CARACTERISTICAS	UNO	DUE	YÚN
Microcontrolador	ATmega328P	AT91SAM3X8E	ATmega32U4
Voltaje operación	5V	3.3V	5V
Voltaje entrada(recomendado)	7-12V	7-12V	5V
Voltaje entrada (limite)	6-20V	6-16V	
Pines digitales	14	54	20
Pines PWM Digitales	6	12	7



Las características por las que se ha hecho muy popular son:

- La mayoría de todos los dispositivos electrónicos en la actualidad tienen instalado de serie.
- Muy reducido el nivel de energía que se le ha de suministrar para su funcionamiento.
- Para el consumidor la utilización del bluetooth es muy sencilla. Con unos pocos pasos la conexión se realiza, se debe conectar el bluetooth de su dispositivo, tocar para que se cree la conexión y esperar a que se inicie la comunicación.

En la siguiente tabla (2) se puede observar una clasificación de los dispositivos bluetooth, esta clasificación es útil si lo que interesa es el alcance al que se va a hacer la comunicación.

Clase	Potencia máxima permitida(mW)	Potencia máxima permitida (dBm)	Alcance aproximado(m)
1	100	20	100
2	2.5	4	5-10
3	1	0	1

Tabla 2 Clasificación bluetooth según el alcance

Otro criterio de clasificación es según su capacidad de canal (tabla3)

Versión	Ancho de banda (Mbit/s)
1.2	1
2.0+EDR	3
3.0+HS	24
4.0	32

Tabla3 Clasificación según capacidad de canal

Las versiones de bluetooth han ido surgiendo a medida que la tecnología iba mejorando, por ejemplo la versión 1.2 que se lanzó al mercado tenía mejoras respecto a sus predecesores como por ejemplo una conexión más rápida y una detección de otros dispositivos bluetooth llamado Discovery, mayor velocidad de transmisión y se introdujo el control de flujo y los modos de retransmisión.

La siguiente versión salió un año más tarde que se caracteriza por la introducción de un nuevo concepto de transmisión de datos más rápido llamado EDR Enhanced Data Rate. La tercera versión siguiendo con la demanda del mercado de hacer dispositivos

con más capacidad de transmisión de datos añadió la posibilidad de usar alternativas MAC y PHY para el transporte de datos. La cuarta y última por el momento de versión, se caracteriza por adoptar la tecnología de bajo consumo. Para conseguirlo se implementó un chip utilizando circuitos CMOS, consisten en la utilización simultánea de transistores de tipo pMOS y nMOS configurados de tal manera que en estado de reposo el consumo de energía es causado por las corrientes parásitas de la placa base a la que conectemos el dispositivo.

### Modulo de bluetooth escogido

En este proyecto la premisa principal es que los elementos que se utilizan ocupen poco espacio y el modulo de bluetooth HC-06 cumple este requisito. Otro punto que se ha fijado es el tema del consumo ya que se quiere un elemento que consuma poco y este modulo lo hace, tiene un consumo de corriente muy bajo de 30 a 40 mA en el modo espera, es decir, modo de estar esperando a que un dispositivo bluetooth solicite la relación y en el modo activo 8mA. Otra parte importante es la parte de la alimentación, se deben utilizar dispositivos que no necesiten mucha energía este dispositivo funciona con 3,3 voltios que corresponde a los voltios que puede general la placa Arduino DUE. Otro punto a favor es la alta fiabilidad de transmisión de datos, es un dispositivo que puedes controlar muchas de sus funciones entre ellas la velocidad de transmisión de datos y por último mencionar que es un dispositivo diseñado para conectarlo a la placa protoboard.

Conviene destacar por ejemplo que es un modulo que guarda en la memoria los dispositivos con los que se ha relacionado, con esta cualidad evita al usuario introducir el código de relación cada vez que se quiere emparejar con el módulo. Este tiene un pin llamado KEY que se encarga de borrar la memoria si se quiere introduciendo corriente. El modulo tiene un LED que indica en qué estado se encuentra puede estar en dos: parpadeo en modo espera, continua, modo activo. De la parte técnica del montaje decir que se debe ir con cuidado con la utilización de los pines ya que el modulo va al revés de la placa Arduino DUE en que el pin Tx del modulo se tiene que relacionar con el pin Rx de la placa y el pin Rx del modulo con el Tx de la placa.

### Información pantalla

Hay dos tipos de pantallas en el mercado las llamadas OLED y las LCD( pantallas de cristal liquido). Estas últimas son pantallas delgadas y planas formadas por un número de pixeles en color o monocromos dispuestos delante de una fuente de luz o reflectora, éstos consisten en una capa de moléculas alineadas entre dos electrodos

transparentes y dos filtros de polarización. En las pantallas de color los pixeles se dividen en tres células de color rojo, verde y azul, cada célula se puede controlar independiente para producir millones de posibles colores para cada pixel.

Las pantallas monocromo son diseñadas con una matriz pasiva, es decir, cada fila o columna de la pantalla tiene un solo circuito eléctrico. Esa formación hace trabajar los pixeles de una manera determinada, son dirigidos a la vez por direcciones de fila y de columna. Este tipo de pantallas emplean una tecnología llamada Super-Twisted Nematic(STN) que consume menos y son más baratas de fabricar o Dual Scan STN(DSTN) que proporcionan una imagen más nítida porque se divide la pantalla en dos mitades y cada una se escanea simultáneamente duplicando el número de líneas refrescadas por segundo. No son convenientes para pantallas más grandes, es decir que la matriz tenga más columnas y filas porque el tiempo de respuesta es lento y con un contraste más pobre. Este tipo de matrices se les llama matrices pasivas.

Las pantallas de color contienen otro tipo de matriz llamada Thin-Film-Transistor que es agregada a la pantalla que polariza y a los filtros de color. Su peculiaridad es que cada pixel tiene su propio transistor que permitirá a cada línea de la columna acceder a un pixel. Si dicha fila es desactivada se activa la siguiente línea de fila, todas las líneas se activan secuencialmente durante un ciclo de trabajo. Esta tecnología está orientada a dispositivos de mayor tamaño, mayor brillo y se busca que tengan un tiempo de respuesta bajo.

El otro gran bloque de tipos de pantallas son las llamadas OLED( Organic Light-Emitting, diodo orgánico de emisión de luz que están compuestas por 2 finas capas orgánicas: capa emisión y capa conducción mas otras dos finas películas que hacen de terminal de ánodo y cátodo. El funcionamiento consiste en la introducción de voltaje de manera que el ánodo sea positivo respecto el cátodo provocando un flujo de electrones en el sentido contrario de cátodo a ánodo. Con este efecto el cátodo cede electrones a la capa de emisión que se carga negativamente por el exceso de electrones y el ánodo los sustrae de la capa de conducción que se carga con huecos. En este punto las fuerzas electrostáticas atraen a los electrones hacia los huecos recombinando las posiciones, ese efecto causa una emisión de radiación a una frecuencia que se encuentra en la zona visible del espectro y esto hace que se observe un punto de luz de un color determinado. La suma de estos puntos de luz que ocurren simultáneamente, lo llamamos imagen.

#### Razones de escoger la pantalla

Se quería una pantalla suficientemente grande para que fuera cómoda para jugar y se viera bien a un precio razonable por esa razón se escogió una pantalla de 4.3" con 40 pines que corresponden a la opción de pantalla táctil, otros que relacionan la pantalla



con la memoria SD, otros que dan la función de pantalla y los canales de buses para dar funcionalidad a la pantalla. Ésta tiene una resolución 480 x 272 este dato será muy importante en el momento de programar ya que define los límites del campo de juego y los extremos hasta donde pondrán moverse las raquetas y servirá para saber situar en la pantalla el marcador. Un dato importante a tener en cuenta es que no tiene una SPI interface como las placas Arduino, en cambio, tiene una interfaz RGB completa que requiere de un procesador potente como los que dota Arduino a sus placas o una tarjeta FPGA(Field Programable Gate Array), que es un dispositivo semiconductor que contiene bloques de lógica que se pueden configurar, para controlar la pantalla.

### **Información zumbador**

Es un transductor electroacústico, es decir, es un dispositivo que transforma la electricidad en sonido o viceversa. Este sonido es continuo o intermitente de un mismo tono. Estos elementos sirven mayoritariamente como alarmas para múltiples sistemas. Tiene un diseño de fabricación sencillo, consta de un electroimán y una lámina metálica de acero, cuando se le introduce corriente pasa por la bobina del electroimán y produce un campo magnético variable que hace vibrar la lamina de acero metálica sobre la armadura produciendo su característico sonido generalmente agudo. El motivo de la utilización de este elemento es para hacer el juego más atractivo y divertido. Y la implementación en el circuito es muy sencilla solo se necesita unirlo con un pin PWM e introducir unas líneas de instrucciones indicando cuando pasar la señal a nivel alto, es decir, cuando quieres que se active.

### **Información placa PCB LCD-DUE**

La introducción de este elemento deriva de la necesidad de reducir el espacio del pack pantalla-arduino y que sea más seguro ya que los cables usados en un principio se pueden soltar del circuito porque están pensados para ser utilizados como cables de pruebas, por esa razón deben ser de fácil desconexión con el circuito. El problema nace en el momento que se piensa en cerrar los componentes en una caja y que solo se vea la pantalla, porque entonces se debe procurar que no se mueva ninguna conexión. La idea de la caja surge como una opción de mejora del proyecto, para hacerlo más interesante ya que pasaría a ser portátil, y por tanto, más fácil de transportar, solo se necesitaría una fuente de alimentación, la caja y un móvil que eso siempre las personas lo llevan encima.

Para esta mejora se ha escogido la placa adaptadora LCD TFT01 Arduino MEGA V2.0, que a pesar que sea pensada para ser utilizada con la placa de Arduino MEGA se ha comprobado que es compatible con la placa Arduino DUE.

## Caja

Para el diseño de la caja se ha utilizado el programa informático de diseño Solidworks y para su construcción la impresora 3D. La caja debe tener un agujero para la fuente de alimentación, otro para la pantalla y otros para los tornillos de sujeción.

## Mando

Para el mando se ha utilizado una función de una página web llamada app inventor2 con la que se puede crear un mando con todas las opciones que se desee y controlar los datos que se envían y reciben. De esta manera se ha hecho más fácil programar y entender cómo funciona la interacción mando (móvil) vía bluetooth con el programa.

## Software

### Programas informáticos para el código del juego

Hay multitud de programas que permiten desarrollar, compilar y simular un código con el objetivo de que la máquina entienda que debe hacer en diferentes condiciones en las que se enfrente. Hay una gran variedad de programas según el nivel del usuario entre éstos tenemos el PSeInt enfocado al aprendizaje del usuario que esta iniciándose en el mundo de la lógica de la programación. Este programa tiene opciones muy útiles como el indicador de seguimiento del hilo de ejecución, una ventana para ver si el proyecto cumple en la realidad con lo deseado. Un dato diferenciador de otros programas es la opción de generar y editar un diagrama de flujo del algoritmo una herramienta básica e imprescindible para el aprendizaje, un punto que facilita la comprensión es su avanzado detector de errores que los señala y describe el problema para que sea fácil encontrar solución.

Una vez el usuario tiene cierto nivel, es decir, está familiarizado con los entornos de programación y tiene claro el concepto de cómo interpreta la máquina las instrucciones puede utilizar una variedad muy extensa de programas de los que los más utilizados son: DEV c++ es un entorno de desarrollo integrado (IDE) para programar en lenguaje C++. IDE es una aplicación informática que proporciona las herramientas necesarias para el desarrollo del software mediante un editor de código de fuente, depurador y ejecutor del código. Otro ejemplo es el llamado Notepad++ es un editor de programación gratuito y de código abierto, puede escribirse en diferentes tipos de lenguaje, tiene aspectos positivos que facilitan su manejo como que puedes abrir más de un programa e ir comparándolos mediante su opción de ver los programas en las diferentes pestañas. Otra función es la de clasificar en colores las instrucciones con el fin de ayudar al usuario a cometer menos fallos. Según las necesidades del usuario escogerá un programa u otro puesto que cada uno tiene sus propias características, otro ejemplo es el llamado code blocks que es un entorno que el propio programa detecta que mejor compilador debe utilizar para que resulte más adecuado para la tarea encomendada. Un lenguaje de programación muy extendido es el Java por esta razón se creó Eclipse que es una plataforma para desarrollar y compilar códigos y posee una interfaz fácil permitiendo al usuario utilizar todas las opciones del programa.

Arduino tiene su propio programa IDE Arduino con él se puede programar, compilar, ejecutar virtualmente o en la realidad conectando una placa Arduino al ordenador.

Este es el programa de desarrollo que se ha utilizado por todas las facilidades y porque al utilizar una placa Arduino es la mejor opción y la más recomendable.

## **Programas informáticos para el código del mando**

Los programas que permiten crear aplicaciones compatibles con Android son diversas entre las cuales hay Mono Android que es un SDK (Kit de Desarrollo de software) que nos permite desarrollar un entorno interactivo que pueda utilizar desde un móvil Android, el lenguaje que utiliza es C es decir un lenguaje común y que todo el mundo puede usar.

Otro programa es ApplInventor que se caracteriza porque el lenguaje que utiliza está basado en la lengua de bloques mucho más intuitiva y rápida de familiarizarse.

Con estos programas el usuario es capaz de crear interfaces que hagan todo lo que quiera, distribuir el espacio a su gusto y dar las acciones que desee a cada elemento del interfaz, es decir posee el control total.

Para este proyecto se ha utilizado ApplInventor porque resulta ser un programa muy sencillo de utilizar con una multitud de posibilidades de crear interfaces. Se vió que era compatible con lo que se quería hacer que es un joystick que se conecte por bluetooth y que enviara los comandos que se quería para luego en el programa escrito en IDE Arduino se pudiera relacionar.

## Descripción del proyecto

### Hardware

En este apartado se quiere dar un punto de vista de los elementos que conforman el proyecto, cómo se relacionan entre si y una explicación de las conexiones utilizadas. En la siguiente figura 1 se muestra el diseño del hardware.

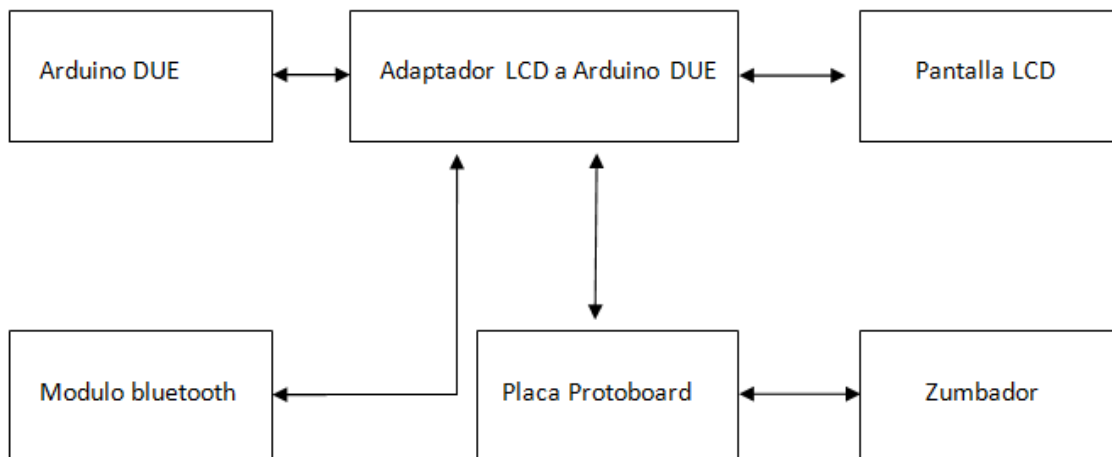


Figura 1 Diseño hardware proyecto

#### Arduino DUE

Este elemento es la parte más importante ya que tiene los pines necesarios para implantar los demás elementos y contiene el cerebro encargado de decir que hacer en cada parte del sistema. Durante gran parte del proyecto ha sido empleado directamente para unir los diferentes módulos electrónicos que se han ido utilizando mediante cables. Esa manera de organizar el conjunto era válida porque no importaba el espacio pero una vez el proyecto estaba muy avanzado se hizo necesario trabajar en paralelo en el diseño y en el desarrollo del programa. Se vio necesaria la reducción de espacio ya que se había utilizado una gran cantidad de cables muy elevada. En ese punto del trabajo se tuvo que buscar algo que supiera los cables, se encontró un adaptador de pantalla LCD a Arduino MEGA que solucionaba el problema.

#### Adaptador LCD-Arduino DUE

El producto encontrado para sustituir los cables es el llamado Shiled LCD Arduino MEGA. Tiene la peculiaridad de conectar los pines necesarios de la placa Arduino con la pantalla LCD en el caso del trabajo de 40 pines. Antes de utilizarla se estudio la composición de la placa Arduino MEGA para ver si era compatible con la placa Arduino

DUE y se observo que era totalmente compatible, por esa razón se pudo utilizar. El adaptador ahora actúa como placa Arduino por esa razón todas las conexiones que se tengan que hacer se harán desde el adaptador. Estas serán en forma de soldaduras para los pines del modulo bluetooth y el conexionado del zumbador, por la otra cara se utilizaran los pines predeterminados tendrán la función de conectarse con la pantalla.

#### Pantalla LCD

La pantalla LCD de 4.3 pulgadas está formada por 40 pines los cuales corresponden a pines que dan funcionalidad a la pantalla táctil, otros activan las funciones de lectura y escritura de la tarjeta SD y el resto activan la pantalla. En el diseño inicial solo se conectaban los pines que dan función a la pantalla para que muestre imágenes mediante cables. En la segunda fase del proyecto se incorporó el adaptador activando todos los pines de la pantalla pero se siguen utilizando solo los pines que activan la pantalla para comunicarse.

#### Modulo bluetooth

Este elemento está conectado mediante unas soldaduras al adaptador, los cables conectados son: el pin VCC que está unido al pin 3,3 V del adaptador, el pin Tx unido al pin Rx3 y el pin Rx al pin Tx3. Este cruce de cables se debe a la configuración del fabricante del modulo bluetooth. Para cerrar el circuito se necesita conectarlo a un GND situado en la placa protoboard.

#### Placa protoboard

Es la encargada de crear puentes entre elementos que no se pueden unir directamente. En este proyecto se utiliza para colocar el zumbador, se coloca un cable que proporciona la energía necesaria para encenderse o apagarse y otro conectado al GND del adaptador. En la misma línea de la placa protoboard se coloca el cable del modulo de bluetooth.

#### Zumbador

Está conectado a la placa protoboard junto al modulo bluetooth. Su función es la de proporcionar un efecto acústico de revote de una pelota de ping pong.



## Software

En la figura 2 se muestra el diagrama de bloques del sistema

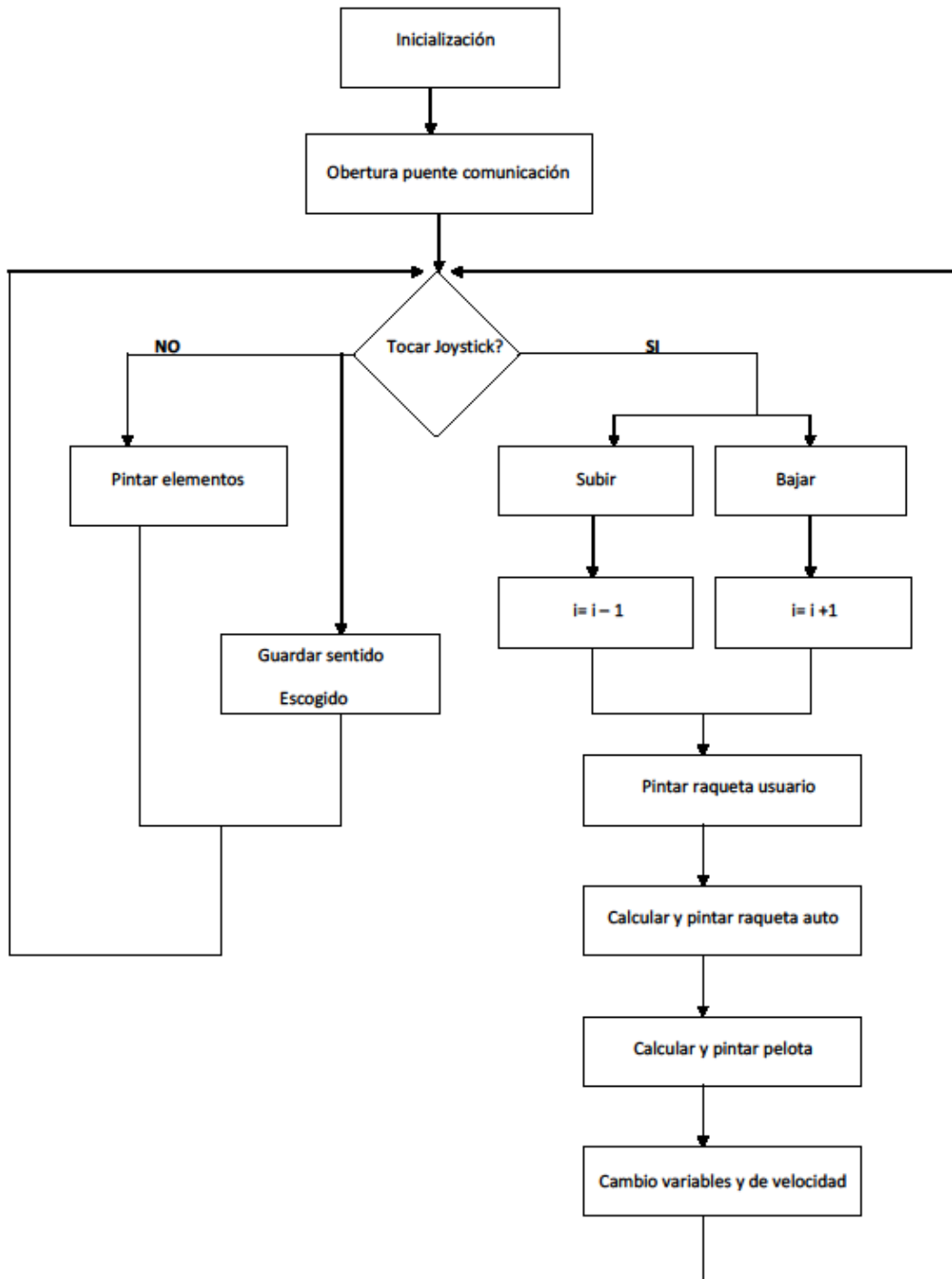


Figura 2 Diagrama de bloques

### **Inicialización**

La primera acción que realiza el programa es la de dar valores iniciales a las variables que se van utilizar, como son las del movimiento de la raqueta controlada por el usuario, las que dirigen a la pelota, las que dirigen a la raqueta automática, variables que cuentan los puntos de cada raqueta. También se han utilizado variables booleanas cuya función es la de detectar si el juego se ha inicializado, detectar el sentido y como debe rebotar la pelota en los extremos de los márgenes de la pista.

### **Obertura puente comunicación**

En este punto se activan las funciones de la placa Arduino para comunicarse con el módulo bluetooth con la siguiente instrucción "Serial.begin(9600);" y se establece la velocidad de transmisión de datos.

### **Tocar joystick**

En esta parte el hilo está constantemente preguntando que quiere hacer el usuario, hay dos posibilidades que son:

- 1- Que el usuario no toque ningún botón, en este caso cabe la posibilidad que el juego se encuentre en dos situaciones: Situación inicial, es decir, cuando se inicia el juego y no se ha tocado ningún botón. En esta situación el sistema pinta los tres elementos que conforman el juego (raquetas, pelota). La otra situación es que el juego ya esté en funcionamiento y lo que hace el programa es guardar la opción de sentido que ha escogido el cliente para mover la raqueta.
- 2- Que el usuario seleccione uno de los dos sentidos en que la raqueta se puede mover: arriba o abajo.

### **i=i+1**

En esta elección el usuario lo que quiere es mover la raqueta hacia abajo. Según el código desarrollado la "i" corresponde a la coordenada Y de la pantalla, se ha sabido que se tenía que sumar por donde está situado el origen de coordenadas. Se posiciona en la esquina superior izquierda.

### **i=i-1**

El usuario selecciona esta opción cuando lo que quiere es desplazar la raqueta hacia arriba, es decir, acercarse al origen de coordenadas.

### **Pintar raqueta usuario**

Llegados a este punto tenemos pintada la raqueta del usuario y se dispone en este bloque a hacerla mover verticalmente, el procedimiento a seguir es mediante la

utilización de una instrucción que nos pinta por pantalla dos rectas, una que borrará un extremo, pintándola de negro y otra recta en el extremo opuesto, de amarillo.

### **Calcular y pintar raqueta automática**

En este proceso se compara la componente “y” de la pelota con la de la raqueta automática según el resultado de la comparación se hará una operación u otra. En el caso de que sea menor la raqueta deberá subir por lo tanto la operación será de disminuir la componente “y” que corresponde a la variable yai. En el caso contrario deberá incrementarse en una unidad y la raqueta bajará.

### **Calcular y pintar pelota**

Esta acción utiliza una instrucción diferente a las demás, es la instrucción drawbitmap que consiste en indicar en qué posición, cuantos pixeles ocupará y mencionar que imagen se quiere imprimir. La coordenada se calcula mediante unas acciones que están asociadas que determinan si se tiene que sumar o restar las componentes “x” o “y” de la circunferencia.

### **Cambio de variables y de velocidad**

Para detectar la posición anterior y poder trabajar con ella se deben añadir unas instrucciones que la función que tienen es la de guardar la posición anterior, éstas se colocan en el último punto del programa.

Una vez ejecutados todos los pasos el hilo del programa llega a las últimas instrucciones que corresponden a un contador de ciclos que determina cuando cambiar de velocidad la raqueta automática con el fin de que vaya más lenta que la pelota para que no llegue a tocarla y de esta forma la maquina falle.

Por último se pinta por pantalla el marcador indicando que punto de las coordenadas de la pantalla se desea que se muestre, se ha escogido un lugar en que no molestara a la hora de jugar.

### **Pintar elementos**

Si el usuario no pulsa ningún botón el juego no se inicia y se queda el programa constantemente preguntando y mientras lo hace pinta por pantalla el marcador a cero, las dos raquetas en sus posiciones iniciales predeterminadas y la pelota en el centro.

### **Guardar sentido escogido**

La diferencia con el anterior estado es que en éste sí que el juego se ha iniciado y en este if la función que tiene es la de memorizar en una variable la opción seleccionada por el usuario para donde querría que fuera la raqueta.

## Resultados

Una vez acabado el proyecto se puede hacer una comparación entre lo que se deseaba en un principio y lo que se ha obtenido. La parte que más ha evolucionado es el mando para controlar la raqueta ya que en sus orígenes se pensaba construirlo desde cero. Es decir, informarse de los elementos que formarían parte del sistema, buscar las mejores opciones entre la multitud de variantes de cada componente, informarse de cómo unir cada parte. Como sería un producto utilizado con las manos, debería tener una forma ergonómica y que no ocupara mucho. Desde los comienzos se observó que habría muchos problemas para escoger elementos adecuados y diseñar el mando para que fuera cómodo de utilizar. Por esas razones se optó por utilizar un mando ya existente, sería una opción válida ya que la creación del mando no era el objetivo de este proyecto. Se escogió un mando de una consola conocida con el fin de encontrar la información necesaria y al escoger un mando hecho se incorporó al proyecto un nuevo concepto que fue el de cambiar la forma de comunicar la información del mando al juego.

Este es otro de los cambios más significativos, con el descarte de la creación del mando, se introdujo la opción de hacerlo inalámbrico adaptándose más a la tecnología actual. Con este avance se conseguía eliminar la dependencia de los cables físicos de comunicación proporcionando más autonomía al usuario para decidir a qué distancia querría jugar. El inconveniente encontrado en este punto fue que no se encontraba información de que datos enviaba cuando alguna tecla era pulsada. Por ese motivo se buscó algún método de utilizar un mando inalámbrico que se sepa que envía, se encontró una página web que se dedica a proporcionar los medios necesarios para que el cliente cree un mando digital para el móvil. La característica fundamental es que entre sus opciones hay la de controlar que información se envía cuando se interactúa con el mando. Con este último se acaba con la evolución durante el proyecto del mando convirtiendo una primera idea de mando analógico a mando digital inalámbrico para el móvil.

El cambio de mando analógico propició un cambio del código comprobando que todo en un sistema si cambia algo afecta al conjunto. Con un mando analógico se pueden leer simultáneamente dos entradas que corresponderían a los dos potenciómetros del mando correspondientes a los dos jugadores, pero en un mando que su camino de comunicación es mediante bluetooth cambia la forma que el programa lee la información de dos entradas, lo que hace más complicado leer en tiempo real los cambios en las dos entradas. Por esta razón se decidió hacer un cambio en el formato del juego y no se jugaría contra otro jugador sino contra la maquina, es decir se cambió

el programa de leer cambios en variables analógicas a variables digitales creando una inteligencia artificial que corresponde a la raqueta dirigida automáticamente.

Un cambio no tan significativo fue el cambio de paleta de colores que se ha utilizado, en sus orígenes la idea principal era utilizar los mismos colores que el juego original. Esos colores se basaban en un fondo negro que simula la pista de juego y los demás elementos que se ven en la pantalla de color blanco, es decir, la pelota, raquetas y el marcador. Al incorporar una pantalla LCD de 4.3 pulgadas en color al proyecto se han podido aprovechar todos los recursos gráficos de que dispone la pantalla para cambiar la tradicional imagen de color blanco y negro por otro tipo de colores que hacen más atractivo el juego, más emocionante, distraído, capta más la atención del jugador y el interés.

## Conclusiones

La realización del trabajo de fin de carrera tiene como fin trasladar todo lo aprendido en la universidad en un ejemplo práctico real. Ese objetivo ha sido alcanzado en cada parte del proyecto ya que en todas se ha hecho necesaria la creación de una planificación horaria y de calendario para el control de los tiempos y llegar a los plazos establecidos, se han seguido las directrices asumidas en las asignaturas impartidas, en ellas los alumnos debían entregar informes con un tiempo limitado de entrega y siguiendo un guión. El cumplimiento de los plazos y el ser consciente del calendario ha contribuido a que el proyecto se terminara en el tiempo establecido.

Otro objetivo cumplido es la manera de buscar la teoría e información de los elementos que componen el sistema. Este conocimiento lo adquiere el alumno en los seminarios impartidos por la escuela, donde se muestra el mejor camino para la obtención de la materia necesaria actualizada y con rigor científico y con la multitud de consejos adquiridos en la realización de los trabajos de las asignaturas. Se hace necesaria la aclaración de que a pesar de que la búsqueda de teoría con los métodos asimilados dio buenos resultados, ya que se encontró la información necesaria, hubo otro tipo de información que resultó más compleja de adquirir ya que correspondía a información interna de los componentes que las empresas no suelen proporcionar pero que para este proyecto se hacía necesaria su obtención.

No tan sólo se han utilizado estos recursos para buscar información de los componentes del sistema sino para encontrar donde comprarlos. La búsqueda ha sido complicada por la multitud de variedad que hay en el mercado de cada componente, se han tenido que buscar las características de todos ellos para descartar e ir reduciendo la lista de opciones posibles y, en todos los casos, la escasez de información por parte del fabricante ha dificultado el proceso.

Otra meta que la universidad desea que los alumnos alcancen es la comprensión de los textos técnicos, científicos, los dibujos técnicos y esquemas electrónicos. A lo largo de todo el proyecto ha sido imprescindible la lectura de textos científicos de muchos campos diferentes para solucionar problemas, encontrar los mejores elementos o saber cómo éstos funcionan. Gracias a esta habilidad se ha podido saber que pines de la placa Arduino utilizar, que pines utilizar de la pantalla, saber cómo funciona el modulo bluetooth, como trabajan todas las partes de los diferentes componentes que forman cada elemento del sistema.

Un alumno al acabar la carrera debe haber asumido una de las cualidades más importantes en un ingeniero que es la de encontrar los errores que hacen que el sistema no actúe como se desea y saber encontrar la solución. En multitud de



ocasiones se han tenido problemas de comunicación con la pantalla, que no se encendía o que no se comunicaba con la placa, de comunicación entre la placa y el modulo bluetooth, de programación en el código del programa, de no saber que enviaba el joystick inalámbrico. Todos esos contratiempos se han podido solventar gracias al análisis de la situación y de enfocar dónde podría estar el problema, procedimientos adquiridos en las prácticas de laboratorio, ya que éstas están orientadas a que los alumnos se enfrenten a dificultades reales, poderlas identificar y resolver.

El trabajo de un ingeniero no está independizado de otras secciones de una empresa por lo tanto debe haber una buena comunicación entre departamentos para que la empresa funcione correctamente. La comunicación que debe tener un ingeniero tiene dos vertientes la escrita y la oral. A lo largo del curso se han tenido que hacer multitud de informes de prácticas, informes de trabajos en grupo en los que se ha aprendido como se debe entregar un texto científico. La otra vertiente es la oral, simultáneamente a los trabajos escritos se han trabajado las presentaciones orales. Se quiere dar relevancia a estos dos tipos de comunicación ya que en una misma empresa el ingeniero tendrá que explicar los avances no tan solo a gente del mismo campo que lo entenderá perfectamente sino a personas de otras secciones que no tiene porque saber los tecnicismos ni la teoría que se aplica. El ingeniero tiene que saber escribir y hablar de forma que todo el mundo entienda lo que se quiere exponer. En este proyecto se elaborará un texto con los objetivos de que sea entendible, fluido pero que contenga toda la información necesaria para que el lector vea que se ha entendido el proyecto. La parte oral será también evaluada, debiendo realizar una presentación del proyecto con el fin de que el público entienda lo que se ha estado desarrollando sin la necesidad de leer el informe.

Una cualidad de todo ingeniero es su trabajo autónomo y el aprendizaje rápido de materia nueva, cualidad que engloba la capacidad para trabajar individualmente siguiendo pautas y fechas límite para la realización de los pequeños objetivos que el propio ingeniero se autodefine. Esta forma de trabajar refleja la fuerza de voluntad de ponerse a trabajar sin que nadie lo diga sólo con el fin de aprender los nuevos conceptos para aplicarlos en resolver el problema. La correcta búsqueda de la teoría, información de uso, facilita que el aprendizaje sea más fácil y rápido, fácil porque si se encuentra el libro donde está correctamente explicado la materia se hace más entendible y si se hace más comprensible y se aprende más rápido.

Como valoraciones personales se ha comprobado que la lectura exhaustiva de los manuales de instrucciones de los componentes, los esquemas que muestran la función de los pines y hacia donde van, los datasheet de los componentes electrónicos o las librerías para saber que instrucciones utilizar y que algoritmos utilizan para saber cómo

trabaja el programa es muy importante para hallar en un primer lugar cómo trabajan y cómo se conectan las partes del sistema y por otra el de buscar errores.

Un sentimiento que ha predominado en todo momento mientras se ha realizado el proyecto ha sido la motivación, gracias en gran parte a la buena distribución del trabajo, de saber dividirlo en problemas mucho más simples, y de esta manera, resolviendo problemas más sencillos cada vez se han ido asumiendo problemas más complejos.

Se ha aprendido a dejar unos días de descanso para desconectar del trabajo y de los problemas que hacen no avanzar en el proyecto, dicha desconexión se puede llevar a cabo porque en la planificación ya se contemplaron ciertos márgenes de días para el reposo, y así cuando se vuelve al trabajo se encuentra la solución más rápidamente. Este recurso está orientado a la parte del desarrollo del programa donde se han encontrado muchas dificultades y a pesar de que se ha trabajado correctamente escribiendo códigos simples e ir aumentando la complejidad ha habido fragmentos del código que han sido difíciles de escribir.

Al finalizar el proyecto de final de carrera se ha podido observar que todo lo asimilado en los cursos se ha puesto en práctica. Principalmente, se ha utilizado todo lo aprendido en las diversas asignaturas de programación, en este grado se nos ha dado una amplia formación en diferentes lenguajes y se han utilizado diversos programas de desarrollo de códigos y así el alumno es capaz de escribir en cualquier lenguaje ya que está familiarizado con los entornos de programación.

Se han utilizado los conocimientos aprendidos en las asignaturas de electrónica, leyendo los manuales o textos electrónicos que explicaban las características y el funcionamiento de los elementos del sistema.

Se ha comprobado también que el mundo real es muy diferente al académico y que la realización de prácticas en los laboratorios, a pesar de que se aproximan a la realidad no se parecen tanto. En la realidad se deben tomar más variables en consideración

Para finalizar, comentar que la realización del proyecto ha sido un trabajo personal que a pesar de las dificultades ha sido muy gratificante realizarlo. Se aprecia que los límites del conocimiento no se han alcanzado y que se puede seguir trabajando en este proyecto después de su entrega.

## Futuras ampliaciones

Por la limitación de tiempo no se han podido hacer varias mejoras que habrían dado más solidez al juego pero que se harán más adelante ya que es factible y se ha ido acrecentando el interés por este juego a medida que se iba avanzado en él.

Una ampliación sería la de aumentar el código para que el juego tenga más opciones de juego como sería modo de entrenamiento, jugar con otro jugador o la selección del nivel de juego, todas ellas opciones muy atractivas de cara al cliente. Son cambios en el código realizables.

Se podría construir un mando, así no utilizar el móvil personal, para ello se debería hacer un estudio de qué elementos se necesitarían y ver como conectarlos. Se conectaría por cables a la pantalla para que se pareciera más a una consola de sala recreativa. Pasar de digital a analógica resultaría fácil ya que supondría cambiar unas pocas instrucciones. El mando constaría de dos botones para cada jugador si lo que se quiere es no cambiar demasiado el código, si por el contrario, lo que se desea es cambiarlo más y que se parezca al PONG original, se cambiaran los botones por potenciómetros, pero en este caso se deberían cambiar muchas más instrucciones, aunque sería perfectamente viable.

En el proyecto se ha construido una estructura para envolver la pantalla y que no se vieran los componentes, que fuera más segura y fija y sobre todo para que se viera un producto ya acabado. Pero en las futuras ampliaciones, con la construcción de un mando se podría construir una nueva estructura que envolviera la pantalla y el mando. Se haría el modelado con el mismo procedimiento utilizando un programa de diseño y imprimiéndolo en 3D.

Una parte importante de los componentes electrónicos es que deben estar conectados a una fuente de alimentación se debería hacer un estudio de viabilidad para incorporar la utilización de alguna otra fuente de energía renovable con el fin de no conectarlo a la red eléctrica.

## Apéndices

### Cronología del desarrollo del programa

Como el programa se basa en la visualización de figuras en una pantalla el primer paso que se debe realizar es la de buscar las instrucciones, en la información de la librería, necesarias para ese objetivo. Se han encontrado multitud de instrucciones para pintar por pantalla un pixel, una recta, círculos, rectángulos o cuadrados, imágenes, textos y números. De cada una de las opciones hay dos variantes, en una solo se pinta el contorno y en la otra se rellena el interior con el color deseado.

Ejemplos de estas instrucciones son:

- `drawPixel(x,y);`
- `drawLine(x1,y1,x2,y2);`
- `drawRect(x1,y1,x2,y2);`
- `drawCircle(x,y,radius);`
- `print(string,x,y);`
- `printNumI(variable,x,y);`
- `drawBitmap(x,y,sx,sy,bitmap);`

En todas las instrucciones la “x” e “y” corresponden al eje de coordenadas de la pantalla, “radius” corresponde al radio del círculo, bitmap al código en c++ que representa una imagen, para realizar el cambio se utiliza el convertidor *ImageConverter565*.

Una vez sabemos cómo imprimir se hacen las instrucciones para que muestre por pantalla lo que envía el mando inalámbrico (móvil), de esta manera nos aseguramos que el código desarrollado del mando está bien hecho y por el otro que las instrucciones del programa son las adecuadas para crear el puente de comunicación vía bluetooth entre el Arduino y el mando. Es de gran utilidad controlar lo que envía el mando para poder hacer correctamente el código, se hace necesario destacar que gracias a este paso se pudo comprobar que el módulo bluetooth siempre está enviando un valor.

El siguiente paso es el de dibujar por pantalla la raqueta, en este punto lo que se desea es entender el funcionamiento de la instrucción y saberla utilizar. Se colocó en un

extremo de la pantalla pensando en la posición final que tendría la raqueta una vez finalizado el juego. La siguiente fase del proceso es la de poder moverla cuando el usuario se lo indique para ello se deben de afrontar dos retos, uno son colocar las instrucciones necesarias para almacenar el valor que envía el modulo bluetooth y el otro redactar el código correctamente para hacer desplazarse la raqueta.

La primera forma de hacer que la raqueta se mueva es mediante la instrucción for que se pone la condición que hasta que no llegue al extremo de la pantalla no pare, solo es necesario modificar una coordenada la correspondiente al eje Y y la del eje X se pone un valor fijo, creando el efecto de desplazamiento. La meta es conseguir que una acción del mando inicie el movimiento de la raqueta por lo tanto vinculamos la variable que representa la altura (eje y) a la variable que representa la información enviada por el modulo bluetooth.

Se observó que un inconveniente era que al enviar el módulo bluetooth una señal fija al juego cuando dejas de pulsar el botón, en lugar de incrementar o disminuir a otro valor vuelve siempre a la señal fija. Con el código realizado hasta este punto la raqueta solo se movía una posición para abajo o para arriba y volvía a la posición que le daba la señal fija. Para solucionarlo se creó una variable que su función era la de guardar el valor del sentido escogido por el usuario con el objetivo de que leyera continuamente el valor escogido y evitar la lectura del fijo del módulo.

El siguiente elemento que se visualiza en el juego es la pelota, se escoge la instrucción deseada y se coloca aproximadamente en el centro con un radio de 8, pero la dificultad que se presenta es hacer mover los dos elementos simultáneamente. En los siguientes intentos se observa como si uno se mueve el otro no se mueve o no se ve, la solución llega en el momento en que se deja de utilizar un for para el movimiento de la raqueta.

Una vez se pueden mover las dos partes se desea crear el movimiento lo más real posible de la pelota, para ello se debe partir la pantalla en cuatro cuadrantes y definir para cada uno una dirección y sentido. En la figura 3 se muestra los cuatro sentidos posibles que puede moverse la pelota.

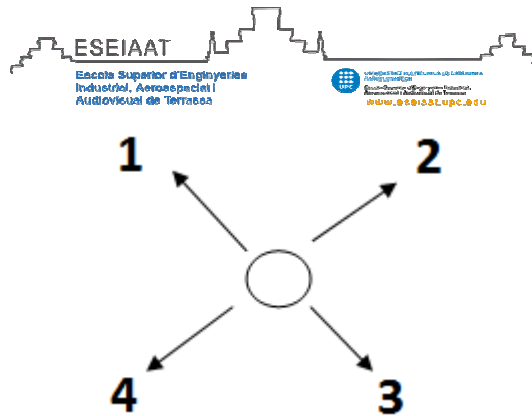


Figura 3 Direcciones pelota

La primera idea a llevar a la práctica fue la de dividir la pantalla en cuatro partes y asignar que cuando la pelota se sitúe en una de esas cuatro partes que se mueva hacia el sentido predefinido. Rápidamente se observó que la pelota no se movía con movimientos aproximados a la realidad por esa razón se tuvo que pensar en otro método que pudiera conseguir movimientos lo más reales posibles. La solución se halló sustituyendo las ecuaciones de las rectas por algo más genérico como son las coordenadas “x” e “y”.

- 1: La componente x disminuye al igual que la y.
- 2: La componente x aumenta y disminuye la y.
- 3: La componente x aumenta al igual que la y.
- 4: La componente x disminuye y aumenta la y.

De esta manera se consigue que el sentido no dependiera de dónde se situaba la pelota sino de dónde viene. Con un algoritmo basado en variables booleanas se ha podido controlar cual de los 4 sentidos tiene que moverse. El juego se basa en la visualización de una pelota rebotando en los márgenes de la pantalla por eso lo que le falta al código son las instrucciones necesarias para que rebote y que lo haga con sentido, simulando la realidad. En la figura 4 se observa el esquema básico del procedimiento que se ha seguido para mover la pelota.

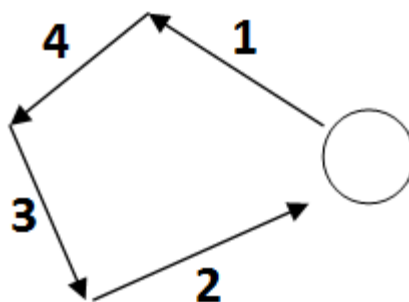


Figura 4 Sentido del rebote de la pelota



El cambio de sentido se debe a dos factores, que la pelota llega al extremo de la pantalla o que es impactada por la raqueta, en la figura 4 se observa como el impacto 1-4 y 3-2 corresponde al límite de la pantalla mientras que el 4-3 significa que la pelota ha impactado contra la raqueta. El impacto de la pelota en los márgenes puede proceder de dos direcciones al igual que el retroceso por esa razón se tuvo que reescribir el código con esa premisa.

Llegados a este punto se quiso mejorar la calidad de imagen ya que el usuario podía ver que los dos componentes de la pantalla parpadeaban. Este problema se solucionó quitando todos los clear screen que no se necesitaban porque había más de los que eran necesarios. Otra modificación fue el de cambiar el orden de las instrucciones para hacer el efecto del movimiento. Se cometió el error de ordenar pintar en la misma ubicación que se daba la de borrar, se corrigió pintando en la nueva coordenada y borrando en la anterior. Estos cambios propiciaron dos efectos en la visualización de los elementos, por un lado se cumplió el objetivo de que no se viera el parpadeo y por otro, la velocidad del juego se incremento en el doble, es decir, la raqueta y la pelota se movían mucho más rápido, eso es debido a que ya no se pintaba y borraba en el mismo sitio.

El comportamiento de la raqueta hasta ese momento era que no se detenía en los extremos de los márgenes, eso dificultaba la forma de jugar ya que cuando desaparecía la raqueta aparecía aleatoriamente en la pantalla. Para solventar el problema se colocaron unos limitadores en forma de "if" que su función era la de detener la raqueta en los dos márgenes.

En este punto se decide añadir el tercer elemento de la pantalla dinámico la raqueta automática, que mide y se mueve con el mismo concepto que la anterior. El reto es saber colocar las instrucciones para que los tres elementos se muevan a la vez, una vez realizado se dispuso a crear la inteligencia artificial que haría la función de mover la raqueta para golpear la pelota. La idea inicial fue la de programar de manera que se fijaría la componente "x" y la coordenada "y" sería la que variaría de la raqueta automática donde la coordenada "y" sería la misma que el valor "y" de la pelota. De esta manera se crea el efecto que solo se mueve en vertical.

Al completar la última tarea el juego básico se terminó con algunos fallos en la raqueta que mueve el usuario y problemas de visualización de los componentes. A partir de este punto los cambios surgidos son para la mejora de la visualización y del rendimiento del programa, es decir hacerlo más eficiente.

Se observó que con la manera de describir los movimientos de la raqueta automática siempre golpeaba la pelota, por lo tanto que nunca perdería. Esto supone un error si lo

que se quiere es hacer un juego real donde ambas palas en algún momento pierdan. Por esa razón se hicieron varios cambios, por un lado se modificó la manera de mover la raqueta que las componentes “y” de la raqueta y la pelota no iban a ser las misma y pasaría a ser que la raqueta perseguiría a la pelota. El otro cambio vino causado por la necesidad que perdiera la maquina lo que propició es que se añadió una variable de velocidad que limitaría la velocidad para que en algunos momento fuera más despacio que la pelota lo cual no llegaría a contactar.

El siguiente paso fue introducir el marcador. En este punto la dificultad estaba en saber qué condiciones poner para que la maquina entendiera que la raqueta no había entrado en contacto con la pelota. Las primeras condiciones no eran suficientes ya que si la pelota golpeaba en una esquina de la raqueta se incrementaba el marcador rival. Para solucionarlo se amplió el rango de zona de rebote.

Una vez colocado el marcador ya se podía hacer dos acciones mas, una en el caso que se gane a la maquina y la otra en el caso de perder. Las dos acciones se basan en la creación de una animación para distraer al jugador.

Los cambios en el código a partir de este punto están orientados a la mejora de la visualización de los tres componentes. Se cambió la manera de mover las raquetas con un concepto que hace más eficiente el código y produce una visión de la raqueta sin parpadeos y se modificó por completo la acción que pintaba la pelota porque se veía con un parpadeo continuo. Se utilizó para ello una imagen de una pelota de tenis solucionando el problema.

## Librería UTFT

La librería que se ha escogido se ha encontrado en la web de un programador Henning Karlsen Fully que se conoció mediante una recomendación que hacia la página web de donde se compró la pantalla. La web se llama Diotronic en que recomendaba a un programador para poder utilizar la pantalla sin problemas. En la web del programador se encontraron multitud de librerías y se escogió la más apropiada a las necesidades del proyecto que fue la UTFT. En la web te describe los pasos para descargarla correctamente y en que carpeta añadirla para que el programa de Arduino lo detecte.

Una vez descomprimida el fichero descargado se añade en la carpeta de librerías del programa de Arduino. En el interior hay una multitud de ficheros los cuales son: la documentación, ejemplos, hardware, licencias, drivers de la pantalla TFT, instrucciones del manejo de imágenes, librería UTFT. Los primeros ficheros que se deben leer están en la carpeta de documentación consta de tres ficheros. Uno son las instrucciones con las que se trabajaran para imprimir por pantalla, otro las instrucciones que se debemos poner en el programa para que conecte la pantalla con la placa y el último el cómo se deben poner los cables para que funcione correctamente.

## Instalar la pantalla con cables

Una vez se tenía la pantalla, los cables, la placa y se supo que librería utilizar se estaba en la disposición de instalar la pantalla. El primer paso fue saber cada pin de la pantalla qué función tenía, en este punto fue complicado encontrar la información ya que en internet se encuentran para otros modelos. El segundo paso fue hacer lo mismo para la placa Arduino DUE pero en este caso fue más sencillo ya que en la web oficial se pudo encontrar toda la información. El tercer paso fue mirar en la librería UTFT la documentación pertinente en este caso al fichero llamado requerimientos. Donde especifica que pines de la pantalla y de la placa se deben unir y cuáles pines van a masa, el voltaje requerido son 3,3V es ese voltaje porque se trabaja con la placa Arduino DUE.

Se encontraron formas de unir pantalla y placa que resultaban eficientes porque se utilizaban pocos pines, siempre conviene que el producto final ocupe menos y cueste menos, pero se tuvieron problemas para llevar a cabo ese procedimiento porque no se utilizaban correctamente los pines. La librería en la que ese procedimiento está relacionada es con la TFT que consiste en declarar 4 variables y utilizar los pines SPI. Esta librería está basada en la extensión Adafruit GFX y la Adafruit ST7735. La librería GFX es responsable de las rutinas de dibujo mientras que la ST7735 es específica para la pantalla TFT en Arduino.

La librería UTFT requiere la utilización de mas pines pero en ésta sí que se supo unir la pantalla y el Arduino porque se encontró la documentación los pines que se debían utilizar para la unión. El material necesario se compone de cables macho/hembra, cables macho/macho, pantalla, un Arduino DUE y una placa protoboard. En este proceso solo se conectaran los pines necesario para que funcione la pantalla sin utilizar las características del SD card o de las opciones táctiles. Los pines que se tienen que relacionar se encuentran en el fichero llamado UTFT requerimientos. En la tabla 4 se ve los pines que están asociados de la pantalla con la placa Arduino DUE:

Pines pantalla	Pines Arduino DUE
DB0	D37
DB1	D36
DB2	D35
DB3	D34
DB4	D33
DB5	D32
DB6	D31
DB7	D30
DB8	D22
DB9	D23

DB10	D24
DB11	D25
DB12	D26
DB13	D27
DB14	D28
DB15	D29
RS	Pin libre
WR	Pin libre
RD	Pin libre
CS	Pin libre
REST	Pin libre

Tabla 4 Relación pines pantalla y Arduino

## Programa comentado

```
#include <UTFT.h> //se incluye la librería UTFT

#define HC06 Serial3// la variable HC06 se une con la salida número 3 del Arduino

extern uint8_t BigFont[];//se define el tamaño de la letra

UTFT myGLCD(TFT01_43,38,39,40,41);// se tiene que especificar una clase de las que hay en la librería, el modelo escogido viene
definido por las componentes del Arduino DUE controlador:SSD1963 corresponde al modelo TFT01_43;los pines
38(RS),39(CS),40(WR),41(RST) son los que indica el fabricante si se utiliza una shield genérica LCD Arduino

extern unsigned short icon[0x100];

int btdata = 0;// variable sentido escogido por el usuario

int btdataint;//variable que guarda el sentido escogido por el usuario

double i=0;// variable que controla el movimiento de la raqueta del usuario, corresponde a la altura

int iant=0;//variable que guarda el ultimo movimiento de la raqueta

double yant=0;// variable que guarda la componente "y" de la circunferencia

double xant=0;//variable que guarda la componente "x" de la circunferencia

bool empezar =false;// variable booleana con función de detectar si los elementos están moviéndose

bool direccion1=false;// variable booleana con función de detectar si la pelota va en el sentido 1

bool direccion2=false;//variable booleana con función de detectar si la pelota va en el sentido 2

bool direccion3=false;//variable booleana con función de detectar si la pelota va en el sentido 3

bool direccion4=false;//variable booleana con función de detectar si la pelota va en el sentido 4

bool gol=false;//variable booleana con función de detectar si la pelota no impacta con la raqueta

bool resultado=false;//variable booleana con función de detectar si se han marcado 5 goles

double x=0;// componente "x" de la circunferencia

double y=1;// componente "y" de la circunferencia

double yai=90;//// componente "y" de la raqueta artificial

double yaiant=0;// componente "y" de la circunferencia anterior

double aispeed =1;//

int ciclos=0;//contador de ciclos del programa

int contador=0;//marcador raqueta usuario

int contadoriz=0;//marcador raqueta automática

int tant=0;//radio circunferencia parte ganar
```

```

int sant=0;//radio circunferencia parte ganar

int mant=0;//radio circunferencia parte ganar

int velocidadpelota=1;//velocidad a la que se mueve la pelota

void setup()
{
  pinMode(49, OUTPUT);// corresponde al pin RD de la pantalla y se define como salida

  pinMode(5, OUTPUT);// corresponde al pin 5 de la placa Arduino y se define como salida

  // Setup the LCD

  myGLCD.InitLCD();//activar características de la pantalla

  myGLCD.setFont(BigFont);//especificas que tamaño de letra

  //Modulo BT

  Serial.begin(9600);//activar las funciones de la placa Arduino referentes a la comunicación bluetooth

  HC06.begin(9600);// activar el bluetooth

  myGLCD.fillScr(0,0,0);// pintar la pantalla de negro
}

void loop()
{
  btdata = HC06.read()-48;// restar -48 a la variable que envía el bluetooth con el fin que trabaje el programa con un 1 o 2

  if (btdata==49 && empezar==false ){ //condiciones del if son cuando no se pulse ningún botón en el mando y no se mueve
ningún elemento de la pantalla

    x=236;//inicialización de la componente "x" de la pelota

    y=100;//inicialización de la componente "y" de la pelota

    direccion1=true; //forzar las direcciones

    direccion2=false;//para que cuando se reinicie el juego vaya por el lado deseado

    direccion3=false;//el juego vaya por el lado deseado

    direccion4=false;

    if(gol==true){ myGLCD.clrScr ();};//si la pelota no entra en contacto con la raqueta se
hace una limpieza de todos los elemntos de la pantalla

    gol=false;

  }

  if (contador==5&&resultado==true){//se entra en el if si se pierde

    perder();//llamamos a la acción perder();
  }
}

```

```

contador=0;//inicializamos los dos marcadores a cero

contadoriz=0;

resultado=false;//forzamos a false así evitamos que entre en este if
    }

if (contador==5&&resultado==true){//se entra en el if si se gana
    ganar();//llamamos a la acción de ganar

    contadoriz=0;

    contador=0;

    resultado=false;
    }

if (contador !=5 &&contadoriz!=5&&resultado==false){//la condición del if es que si no se llega a los 5 puntos entra
en el if

    myGLCD.fillRect(1, (0+i) , 10, (30+i));// pinta rectángulo usuario

    myGLCD.drawBitmap (x, y, 16, 16, icon);

    myGLCD.setColor(255,255,0);

    myGLCD.fillRect(469, (yai-15) , 479, (yai+15) );//pinta raqueta automática
    }

}

if (btdata==49 && empezar==true ){ //este if hace la función de que si hay elementos que se mueven y no se pulsa ningún
botón en el mando mueve la raqueta automática, la pelota y guarda la variable escogido con anterioridad el usuario

    btdata=btdata+1;

    moveraquetaauto();

    }

if (btdata==1) {//si el usuario escoge subir la raqueta

    if (i>0) {

        i=i-1;// se resta 1 a la componente "y" de la raqueta del usuario

        mover();//se llama a la acción mover

        moveraquetaauto();//se llama a la acción moveraquetaauto

        }

    pelota();//se llama a la acción pelota

}

```



```
if (btdata==2) { //si el usuario escoge bajar la raqueta
```

```
    if (i<242) {
```

```
        i=i+1; // se suma 1 a la componente "y" de la raqueta del usuario
```

```
        mover();
```

```
        moveraquetaauto();
```

```
    }
```

```
    pelota();
```

```
}
```

```
digitalWrite(5, LOW); //se pone a señal baja la salida 5
```

```
iant=i; //se guarda la variable i actual en la variable i anterior
```

```
yant=y; // se guarda la componente "y" del punto central de la circunferencia en la variable yant
```

```
xant=x; // se guarda la componente "x" del punto central de la circunferencia en la variable xant
```

```
ciclos++; // se incrementa en 1 el valor de ciclos
```

```
if (ciclos==200){ // condición para que entre en el if es que los ciclos sean igual a 500
```

```
    aispeed= random ( 50,100);
```

```
    aispeed=aispeed*0.01;
```

```
    ciclos=0;
```

```
}
```

```
myGLCD.printNumI(contadoriz, 200, 20); // se indica que pinte el valor del contador y su posición respecto el eje de  
cordenadas
```

```
myGLCD.printNumI(contador, 250, 20); // se indica que pinte el valor del contador y su posición respecto el eje de  
cordenadas
```

```
delay(5); // se introduce un parón en el programa para evitar parpadeos en la visualización de los elementos
```

```
}
```

```
void mover(){ // accion que pinta la raqueta del usuario
```

```
    empezar=true; // forzamos empezar a true
```

```
    // hay dos if se diferencia en este punto si el usuario ha tocado arriba(1) o abajo(2)
```

```
    if (btdata==1){
```

```
        myGLCD.setColor(255,255,0);
```

```
        myGLCD.drawLine(1,(0+i),10,(0+i)); //pinta en la posición anterior en amarillo
```

```
        myGLCD.setColor(0,0,0);
```

```

myGLCD.drawLine(1,(30+iant),10,(30+iant)); // pinta en negro la posición anterior
    }

    if (btdata==2){

        myGLCD.setColor(0,0,0);

        myGLCD.drawLine(1,(0+iant),10,(0+iant));

        myGLCD.setColor(255,255,0);

        myGLCD.drawLine(1,(30+i),10,(30+i));

    }

    btdata=btdata;

}

void M1(){

    x=x-1;//componente "x" de la pelota se decrementa en una unidad

    y=y-1;//componente "y" de la pelota se decrementa en una unidad

    //hay dos direcciones posibles de rebote en el caso que la pelota vaya en dirección 1

    if (y<8 && x> 20){

        direccion1=false;// es el limitador de los márgenes de la pantalla, es decir indicar en qué punto
rebotar la pelota

        direccion4=true;// se fuerza a que vaya hacia la dirección correcta

        digitalWrite(5, HIGH);//se eleva la señal a lo mas alto para activar el zumbador

    }

    if(x<18 && direccion1==true){

        direccion1=false;

        direccion2=true;

        digitalWrite(5, HIGH);

        contador();

    }

}

void M4(){

    x=x-1;//componente "x" de la pelota se decrementa en una unidad

    y=y+1;//componente "y" de la pelota se incrementa en una unidad

    //hay dos direcciones posibles de rebote en el caso que la pelota vaya en dirección 4

    if (x<18 && direccion4==true){

```

```

    direccion4=false;

    direccion3=true;

    digitalWrite(5, HIGH);

    contadord();

}

if (y>272 && direccion4==true){

    direccion4=false;

    direccion1=true;

    digitalWrite(5, HIGH);

}

}

void M2(){

    x=x+1;//componente "x" de la pelota se incrementa en una unidad

    y=y-1;//componente "y" de la pelota se decrementa en una unidad

    //hay dos direcciones posibles de rebote en el caso que la pelota vaya en dirección 2

    if (x>450){

        direccion2=false;

        direccion1=true;

        digitalWrite(5, HIGH);

        contadori();

    }

    if (y<8){

        direccion2=false;

        direccion3=true;

        digitalWrite(5, HIGH);

    }

}

void M3(){

    x=x+1;//componente "x" de la pelota se incrementa en una unidad

    y=y+1;//componente "y" de la pelota se incrementa en una unidad

    //hay dos direcciones posibles de rebote en el caso que la pelota vaya en dirección 3

```

```
if(y>272&& direccion3==true){
```

```
    direccion3=false;
```

```
    direccion2=true;
```

```
    digitalWrite(5, HIGH);
```

```
}
```

```
if (x>450&&direccion3==true ){
```

```
    direccion3=false;
```

```
    direccion4=true;
```

```
    digitalWrite(5, HIGH);
```

```
    contadori();
```

```
}
```

```
}
```

```
void moveraquetaaauto(){//acci3n que determina el sentidu en que se tiene que mover la raqueta autom3tica
```

```
    if (y>=yai){ //baja. compara la componente "y" de la raqueta autom3tica con la componente "y" de la pelota
```

```
        yaiant=yai;
```

```
        myGLCD.setColor(0,0,0);
```

```
        myGLCD.drawLine(469, (yaiant-15) , 479, (yaiant-15));
```

```
        yai=yai+(aispeed);
```

```
        myGLCD.setColor(255,255,0);
```

```
        myGLCD.drawLine(469, (yai+15) , 479, (yai+15) );
```

```
    }
```

```
if (y<yai){ //sube la raqueta autonamitca
```

```
    yaiant=yai;
```

```
    myGLCD.setColor(0,0,0);
```

```
    myGLCD.drawLine(469, (yaiant+15) , 479, (yaiant+15) );
```

```
    yai=yai-(aispeed);
```

```
    myGLCD.setColor(255,255,0);
```

```
    myGLCD.drawLine(469, (yai-15) , 479, (yai-15));
```

```
}
```

```
}
```

```
void contadord(){
```

//los if definen el margen el cual si se cumple la condición considera el programa que no se ha dado a la pelota y sube en el marcador

```
    if (y < (i-7)){
```

la raqueta usuario

```
        contador++; // "y" corresponde la a coordenada "y" de la pelota , la "i" corresponde a la "y" de
```

```
        empezar=false; //forzamos a entrar en el primer if del main
```

```
        gol=true; // detectamos que se ha marcado
```

```
        if (contador==5){resultado=true;}
```

```
    }
```

```
    if (y > ((30+i)+7)){
```

```
        contador++;
```

```
        empezar=false;
```

```
        gol=true;
```

```
        if (contador==5){resultado=true;}
```

```
    }
```

```
}
```

```
void contadori(){
```

```
    if (y < ((yai-15)-7)){
```

raqueta automatica

```
        contadoriz++; // "y" corresponde la a coordenada "y" de la pelota , la "yai" corresponde a la "y" de la
```

```
        empezar=false;
```

```
        gol=true;
```

```
        if (contadoriz==5){resultado=true;}
```

```
    }
```

```
    if (y > ((yai+15)+7)){
```

```
        contadoriz++;
```

```
        empezar=false;
```

```
        gol=true;
```

```
        if (contadoriz==5){resultado=true;}
```

```
    }
```

```
}
```

```
void pelota(){
```

```
// se define la siguiente posición que debe tener la pelota
```

```
if (direccion1==true &&direccion2==false && direccion3==false &&direccion4==false){M1();} //se fuerza para que  
vaya en sentido 1
```

```
if (direccion1==false &&direccion2==false && direccion3==false &&direccion4==true){M4();} //se fuerza para que  
vaya en sentido 4
```

```
if (direccion1==false &&direccion2==false && direccion3==true &&direccion4==false){M3();} //se fuerza para que  
vaya en sentido 3
```

```
if (direccion1==false &&direccion2==true && direccion3==false &&direccion4==false){M2();} //se fuerza para que  
vaya en sentido 2
```

```
myGLCD.drawBitmap (x, y, 16, 16, icon);
```

```
}
```

```
void perder (){//acción que se activa cuando el usuario pierde
```

```
for (int t=0; t<=450;t++){// se define un for para crear un efecto de una línea que se mueve mediante  
representación de píxeles
```

```
myGLCD.setColor(255,255,0);
```

```
myGLCD.drawPixel(t,150); // se deja la componente "y" constante, se modifica el  
valor de "x"
```

```
delay (10); // se hace esperar para que no dibuje tan rápido
```

```
digitalWrite(5, HIGH);
```

```
if(t==100){
```

```
digitalWrite(5, LOW); // cuando la variable t llega a 100 se desactiva  
el zumbador y se pinta una recta vertical
```

```
delay(50);
```

```
myGLCD.drawLine(100,150,100,50);
```

```
digitalWrite(5, HIGH);
```

```
delay (1000);
```

```
digitalWrite(5, LOW);
```

```
}
```

```
if(t==200){
```

```
digitalWrite(5, LOW);
```

```
delay(80);
```

```

myGLCD.drawLine(200,150,200,50);

digitalWrite(5, HIGH);

delay (1000);

digitalWrite(5, LOW);

}

} //cerrar for

myGLCD.print("has perdido", 250, 100); // se imprime un texto por pantalla al terminar la animación

delay (1000);

digitalWrite(5, LOW);

myGLCD.fillScr(0,0,0);

empezar==false ;

}

void ganar(){ // se entra cuando el usuario a ganado a la maquina

    for (int s=3;s<30;s++){ // la función del for es la de crear un efecto de fuegos artificiales relacionado la variable s con el radio de
    las circunferencias

        // se dibujan en varios sitios las circunferencias

        myGLCD.setColor(0,0,0);

        myGLCD.drawCircle(100,60,sant);

        myGLCD.setColor(255,255,0);

        myGLCD.drawCircle(100,60,s);

        myGLCD.setColor(0,0,0);

        myGLCD.drawCircle(418,212,sant);

        myGLCD.setColor(255,255,0);

        myGLCD.drawCircle(418,212,s);

        myGLCD.setColor(0,0,0);

        myGLCD.drawCircle(230,50,sant);

        myGLCD.setColor(255,255,0);

        myGLCD.drawCircle(230,50,s);
    }
}

```

```

    delay(100);

    sant=s;//se guarda el radio presente en una variable encargada de identificar que circunferencia borrar
}

myGLCD.fillScr(0,0,0);

digitalWrite(5, HIGH);

for (int m=3;m<30;m++){

    digitalWrite(5, LOW);

    myGLCD.setColor(0,0,0);

    myGLCD.drawCircle(10,200,mant);

    myGLCD.setColor(255,255,0);

    myGLCD.drawCircle(10,200,m);

    myGLCD.setColor(0,0,0);

    myGLCD.drawCircle(400,12,mant);

    myGLCD.setColor(255,255,0);

    myGLCD.drawCircle(400,12,m);

    mant=m;

}

digitalWrite(5, HIGH);

myGLCD.print("has ganado", 250, 150);

delay (1000);

digitalWrite(5, LOW);

myGLCD.fillScr(0,0,0);

empezar==false ;

}

```



## Programa comentado mando

En este apartado se comentará las instrucciones que se han utilizado para dar función al mando, el lenguaje utilizado es el llamado por bloques, éste ha facilitado la escritura ya que es muy intuitivo. En la figura 5 se muestra la interfaz que el usuario tiene para colocar todo lo que necesite para su prototipo de mando y en la figura 6 se muestran las instrucciones que se han utilizado, como se puede observar no se han necesitado demasiadas.

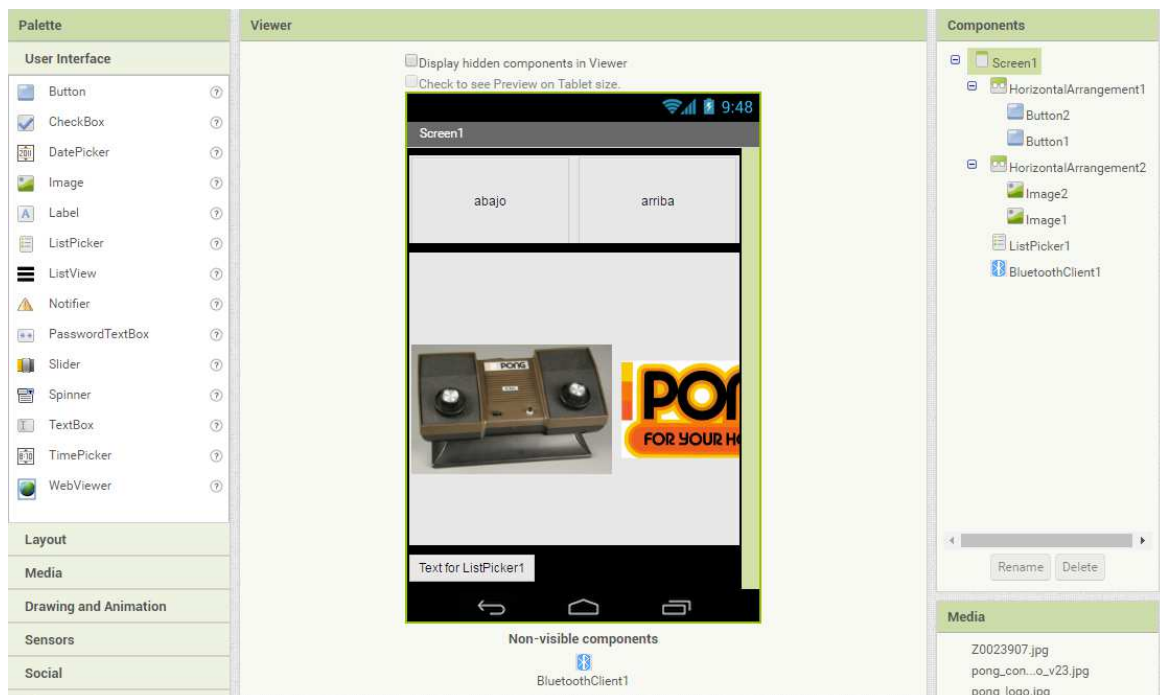


Figura 5 Interfaz programa Appinventor

Se observa a la izquierda de la figura 5 una lista de opciones que se pueden añadir a la pantalla del mando también hay una gran variedad de alternativas de conectividad con los diversos medios sociales.

```

when ListPicker1 .BeforePicking
do
  if BluetoothClient1 . Available
  then set ListPicker1 . Elements to BluetoothClient1 . AddressesAndNames

when ListPicker1 .AfterPicking
do
  evaluate but ignore result call BluetoothClient1 .Connect
  address ListPicker1 . Selection

when Button2 .Click
do
  call BluetoothClient1 .SendText
  text " 2 "

when Button1 .Click
do
  call BluetoothClient1 .SendText
  text " 1 "
  
```

Figura 6 Código mando

En la figura6 se observa qué valores se envían cuando algún botón es pulsado y como hace la máquina para conectarse vía bluetooth.

## Presupuesto material

En la siguiente tabla 5 se muestra el valor de cada uno de los elementos que conforman el producto y el precio total que ha costado.

Material	Euros
Placa Arduino DUE	14
Pantalla TFT LCD SCREEN MODULE 4.3 pulgadas	50,82
Shield TFT01 LCD Arduino MEGA V2.0	11,42
Placa protoboard	5,45
Zumbador	1,46
Cables	1,50
Módulo bluetooth hc-06	13
Caja 3D	13
Tornillos, tuercas	0,80
<b>TOTAL</b>	<b>111,45</b>

Tabla 5 Precio elementos producto

## Bibliografía

Páginas web:

- Arduino: información comunicaciones

<https://www.arduino.cc/en/Reference/SPI>

Día consulta: 11 de marzo

- Arduino: mapa pines SAM3X

<https://www.arduino.cc/en/Hacking/PinMappingSAM3X>

Día consulta: 19 de marzo

- Arduino: lenguaje

<https://www.arduino.cc/en/Guide/TFTtoBoards>

<https://www.arduino.cc/en/Reference/TFTLibrary>

<https://www.arduino.cc/en/Guide/TFT#toc8>

Día consulta: 9 de abril

- Atmel

[http://www.atmel.com/Images/Atmel-11057-32-bit-Cortex-M3-Microcontroller-SAM3X-SAM3A\\_Datasheet.pdf](http://www.atmel.com/Images/Atmel-11057-32-bit-Cortex-M3-Microcontroller-SAM3X-SAM3A_Datasheet.pdf)

Día de consulta: 18 de marzo

- Olimex: información módulo bluetooth

<https://www.olimex.com/Products/Components/RF/BLUETOOTH-SERIAL-HC-06/resources/hc06.pdf>

Día consulta: 19 de marzo

- Prometec: información modulo bluetooth

<http://www.prometec.net/bt-hc06/>

Día consulta: 18 de marzo

- Bluetooth: teoría

<https://www.bluetooth.com/what-is-bluetooth-technology/bluetooth>

<http://www.vs.inf.ethz.ch/publ/papers/Internet-of-things.pdf>

<https://www.cognizant.com/InsightsWhitepapers/Reaping-the-Benefits-of-the-Internet-of-Things.pdf>

<http://web.archive.org/web/20110308041251/http://knol.google.com/k/max-iskram/digital-electronic-design-for-beginners/1f4zs8p9zgq0e/22>

Día consulta: 28 de abril

- Información pantalla

[http://www.elec Freaks.com/wiki/index.php?title=4.3%22\\_Width\\_480\\*272\\_TFT\\_LCD\\_Module](http://www.elec Freaks.com/wiki/index.php?title=4.3%22_Width_480*272_TFT_LCD_Module)

[http://www.coldtears-electronics.com/images/Shield\\_pinout.jpg](http://www.coldtears-electronics.com/images/Shield_pinout.jpg)

<http://hipertextual.com/archivo/2013/11/tipos-pantalla-led-lcd-ips-oled/>

<http://www.pacificdisplay.com/technical.htm>

[https://es.wikipedia.org/wiki/Pantalla\\_de\\_cristal\\_l%C3%ADquido](https://es.wikipedia.org/wiki/Pantalla_de_cristal_l%C3%ADquido)

<https://www.adafruit.com/products/1591>

<http://www.jayconsystems.com/4-3-40-pin-tft-display-480x272-with-touchscreen.html>

Día consulta: 31 de marzo

- Información zumbador

<https://es.wikipedia.org/wiki/Zumbador>

Día consulta: 30 de abril

- Librerías UTF

<http://www.rinkydinkelectronics.com/library.php?id=51>

Día consultado: 16 de abril

- Números ASCII

<http://ascii.cl/es/>

Día consulta: 16 de abril

- joystick Arduino

<http://kio4.com/appinventor/9bluetootharduino.htm>

Día consulta: 30 de abril

- Elecbreaks: información placa adaptador

[http://www.elecbreaks.com/wiki/index.php?title=TFT01\\_Arduino\\_Mega\\_LCD\\_Shield](http://www.elecbreaks.com/wiki/index.php?title=TFT01_Arduino_Mega_LCD_Shield)

<http://www.elecbreaks.com/estore/lcd-tft01-arduino-mega-shield-v2-0-shd10.html>

Día consulta: 2 de abril

- informacion pong

<http://www.atarimuseum.com/videogames/dedicated/homepong.html>

Día consulta: 22 de mayo